Introduction

> In order to be in a relation with the world informatically, one must erase the world, subjecting it to various forms of manipulation, preemption, modeling, and synthetic transformation. . . . The promise is not one of revealing something as it is, but in simulating a thing so effectively that "what it is" becomes less and less necessary to speak about, not because it is gone for good, but because we have perfected a language *for* it.
>
> —Alexander Galloway, *The Interface Effect*

> Doing with images makes symbols.
>
> —Alan Kay, *Doing with Images Makes Symbols*

In the fall of 1972, Marsha Sutherland spent several weeks driving around Salt Lake City in a Volkswagen Beetle half covered in a polygon mesh. The car was a spectacle, its green exterior dotted with hundreds of numbered vertices connected to form a grid of irregular squares (figure 0.1 and plate 1).[1] Marsha had moved to Salt Lake from Cambridge, Massachusetts, just four years prior with her husband, Ivan, who left Harvard University in 1968 for a tenured position in the computer science program at the University of Utah. Each week Marsha would drive up the foothills of the Salt Lake Valley to the Merrill Engineering Building, where Ivan's students would carefully mark and measure the car for digitization. Along the way she would traverse a grid of a different sort: the lockstep raster of city blocks that make up the Plat of Zion, the plan for a city of God first devised by Joseph Smith in 1833, and dug out of the valley floor by Brigham Young and his followers with the colonial settlement of Salt Lake City in 1847.[2] By the end of the year, Marsha's Beetle would become the first real-world object to be fully scanned and rendered by a computer—the physical made digital (figure 0.2).[3]

a)

b)

c)

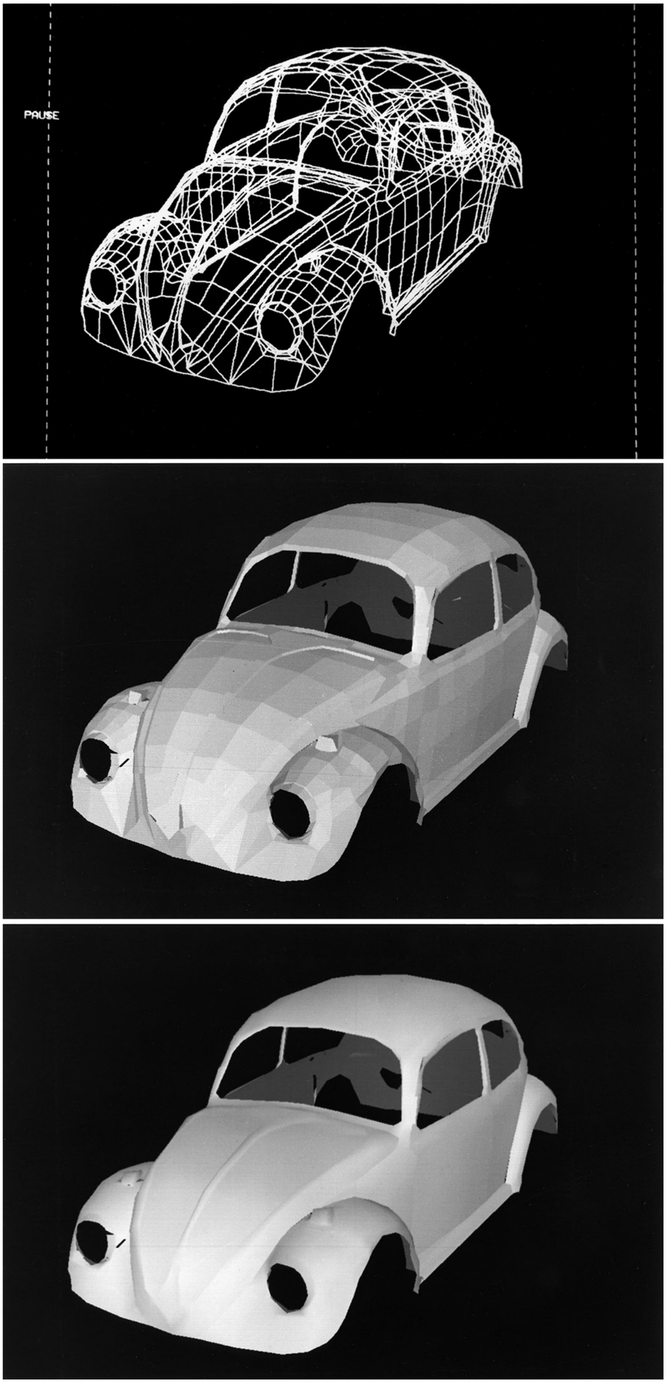

Figure 0.2
Digitized Volkswagen Beetle rendered in wireframe, flat shading, and smooth shading at the University of Utah computer graphics lab, 1973. Courtesy of the Computer History Museum and the University of Utah School of Computing.

A surprising object in an unlikely place, Marsha's Volkswagen straddles two worlds. A global symbol of 1960s' counterculture, the Beetle was near ubiquitous at the start of the 1970s. Earlier that year, in February 1972, the Beetle surpassed the Ford Model T to become the most widely manufactured vehicle ever produced, its design largely unchanged since 1938.[4] It was this iconic status that drew Ivan's students to it and made it legible as an object for simulation in the first place.[5] Yet this particular Beetle marks the beginning of a radical transformation in the shape of our lived environment—a turning point in which the physical world becomes saturated with digital objects. Think, for a moment, of the building in which you now sit, the phone in your pocket, the book you now read; each of these objects have been materially shaped by a process that can be traced to Marsha, winding her way up the hills outside Salt Lake City half a century ago. Each of these objects have, over the course of their design and creation, been touched and transformed by computer graphics.

This may be surprising to anyone accustomed to thinking of graphics exclusively as visual images produced, augmented, or transformed by computation. Likewise, for most of us computer graphics are a relatively recent invention, emerging at the end of the twentieth century as spectacular visual effects and lifelike simulations in film, television, and digital games. In fact, computer graphics are as old as the modern computer itself, and their development marks a fundamental transformation not only in the way we make images, but in the way we mediate our world through the computer, and in turn come to reimagine the world as computational. We live in a world that has been structured by the visual regime of computer graphics. Whether captured with a digital camera, designed and rendered using 3D interactive software, or simply displayed on the pixelated grid of a computer screen, almost all images we view, make, and interact with on a daily basis are shaped by computation. Yet computer graphics have largely disappeared as a legible object of analysis, and the history of computer graphics remains almost entirely unwritten.[6]

This is due in part to the phenomenal invisibility of computer graphics as a distinct technical medium. Most computational images we encounter are designed to simulate and reproduce the formal and aesthetic norms of those media that precede them, be it the photo-realistic renders of special effects and digital games, or the skeuomorphic interfaces of our laptops and smartphones. Consequently the more advanced computer graphics become,

the less visible they appear to be and the less we remark on their ubiquity.[7] When graphics do register as objects for critique, they are almost always framed by discourses of realism and mimesis, or broad narratives of technological development that lead inevitably toward verisimilitude.[8] Computer graphics are perhaps the only medium that is analyzed exclusively in terms of the ways it successfully produces its own invisibility. We might value and remark on the photographic, televisual, or cinematic quality of a media text, but if an image reads as computer graphics, it has failed its simulation. This is because unlike those media that claim an indexical relationship to the world they represent, the thing reproduced by computer graphics is not the world but another medium in simulation. Computer graphics are thus always already mediated, and the goal of nearly all graphical research is the accurate reproduction of the effects of this prior mediation. This mimetic quality has precluded an examination of computer graphics that takes seriously its historical emergence as a distinctly computational technology untethered from the long history of visual representation. Likewise, it has limited our engagement with computer graphics to only their most visible manifestations: as images on screens.

This book begins with the premise that computer graphics are much more than the images we see. They are one of the principal technologies of our historical present, and have reshaped the way we understand, relate with, and engage the material world today. To understand this transformation will require a material and local history of computer graphics as it developed alongside the modern computer in the second half of the twentieth century. Taking up this task, *Image Objects* traces the history of computer graphics in the thirty years prior to the technology's emergence in popular visual culture. In this it offers two interrelated stories.

First this is a history of the computational image, and those technologies that made possible its appearance on the experimental screens of academic and commercial research centers some sixty years ago. Refusing popular narratives of convergence and remediation, I argue that computer graphics is a unique medium distinct from those earlier visual forms it seeks to simulate. To understand and make visible the material specificity of computer graphics, I pull apart the rendered image and identify its constitutive parts: those historical objects that make up the material history of graphical simulation, and through which we might posit a theory of graphical computing. To this end, I ask not simply how computer graphics developed over the second

half of the twentieth century, or who helped shape the discipline through research and innovation, but rather what historical technologies structured and limited the field as it evolved, and how those technologies continue to determine the ways we engage with computational images today. In each of the following chapters I dig out a single technical object, broadly construed, that in turn becomes emblematic of an entire practice of image making. Through an analysis of these five objects that shaped the early history of computer graphics—an algorithm, an interface, an object standard, a programming paradigm, and a hardware platform—*Image Objects* reflects on the ways that visibility, memory, simulation, relation, and history are each inscribed into the technical infrastructure of the medium of computer graphics itself. In turn these objects form the basis for my broadly materialist methodology: an archaeology of this seemingly immaterial media form, the computational image.

I adopt this term strategically as a means of signaling a set of political and methodological concerns, and as part of an effort to place this text in dialogue with a broad field of practice. Both a theory and method, media archaeology encapsulates a great number of media historical interventions. What draws me to this field is, following Vivian Sobchack, its concern for the materiality of media *objects* over the linear teleology of "realist historical representation, which attempts to fill in the absences of the past with coherent—and metaphorical—narratives that substitute for their loss."[9] Media archaeology excavates dead media objects and brings them to bear on the present through a descriptive contextualization that is concerned primarily with what an object is and how it functioned rather than what it might have been interpreted to mean. That said, media archaeology is not exclusively concerned with old and dead things, the forgotten practices associated with them, and their impact on the coproduction of past knowledge. It is also concerned with media as a mode of engaging the material world, and the ways that media act as sensory prostheses that mediate practice and experience. The primary distinction here is that of materiality over representation, and a critique of progressivist, revolutionary, and linear forms of history. I view this book as an archaeology because it looks to a neglected prehistory that has been assumed or obscured by popular discourses of graphical realism. Likewise, in focusing on a series of objects that—while deeply important to the historical function of graphics—have been largely forgotten or taken as given by contemporary researchers, this

project points to the dead media of existing digital forms. Holding in tension the need for a cultural politics of technology and the desire to deprivilege human-centered narratives of technological innovation, I acknowledge the difficulty of describing and performing a fixed media archaeological method, and view it as an essential focus of my investigation.[10]

Further complicating those historical narratives that would presume the centrality of the sites and objects that dominate our contemporary media landscape, *Image Objects* frames the history of computer graphics through a unique but largely neglected site in the history of computing. At a time when the vast majority of computational research was concentrated at university and corporate research institutions on the East and West Coasts, the field of computer graphics developed largely at secondary sites that have been left out of the broader history of computing.[11] Chief among these is the research program at the University of Utah, founded in 1965 by Salt Lake City native David C. Evans and heavily funded by the Department of Defense with the goal of advancing research into "graphical man-machine communication."[12] In the period from roughly 1965 to 1980, the faculty and graduates of the Utah program were responsible for no less than inventing the very concepts that make modern computer graphics possible, and many of the school's graduates went on to become industry leaders in the field of computing in the second half of the twentieth century.[13] The founders of Pixar, Adobe, Silicon Graphics, Netscape, Atari, and WordPerfect were all students at Utah during this period, and dozens of key researchers at Xerox PARC, NASA's Jet Propulsion Laboratory, the New York Institute of Technology, and Industrial Light & Magic all began their careers in Salt Lake City. The University of Utah was the epicenter of graphical development for the first fifteen years of the discipline, and its archives and papers form the foundation of this book. Grounding the history of computer graphics in this way—at a particular historical site and through a discrete set of technologies—*Image Objects* extends a theory of computer graphics not as an ephemeral abstraction but as a physical thing: the digital image as material object.

In tracing the history of computer graphics in this way, *Image Objects* tells a second story about the emergence of a new object form, and along with it the transformation of computation as a technical and cultural practice. Prior to the 1960s, computers were machines built for the procedural calculation of numerical data. They functioned hierarchically, with large mainframes designed for solving predetermined problems or processing

data according to predetermined procedures. Computing was an explicitly noninteractive process; its inputs and outputs were punch cards and paper, and its objects were logic and numbers. Computer graphics was first developed as a means of abstracting computational processes toward human readable modes of interaction—that is, of bringing the material logic of the sensible world to bear on the informatic logic of computational systems. Through computer graphics the image world was operationalized, made to compute and perform actions, to take up and simulate space. The development of computer graphics in this sense marks a reorientation of computer science toward the object world such that it could be made subject to computational forms of simulation, transforming the computer from a *tool* for procedural calculation into a *medium* structured by a distinct ontological claim. Over the past fifty years, this claim has become one of the dominant modes of engaging with and thinking through all manner of processes, such that our contemporary world is now populated by a vast number of objects shaped by their encounter with graphical systems—that is, *image objects*.

The image object here marks a theory and method for engaging the transformation of the visible world under computation. It insists first that digital images are materially structured by those historical objects that produce them—objects that have been rendered out of the visible image, but that fundamentally shape the function and appearance of computer graphics as a distinctly computational technology. From microprocessors and graphics libraries to software suites and shading algorithms, computer graphics contain a vast number of objects whose material histories are erased if we restrict our analysis to the rendered image alone. At the same time, the image object affirms the broad influence of computer images on the shape and function of the material world today, describing the historical process whereby a vast number of material objects have been taken up by computer graphics and made subject to the logic of the digital image. Over the course of this history we will find countless objects taken up and transformed in this way. From the shape of contemporary architecture and built environments, the aesthetics of digital printing and desktop publishing, the interfaces we use to engage and communicate with our world, industrial design and rapid manufacturing, the structure of cars, planes, and other vehicles, and even the design of chips, circuits, and computer hardware itself—all are mediated and informed by this dual logic: at once visual and material, representation and calculation, both image and object.

It can be difficult to see the influence of computer graphics on our lived environment. The processes that define and articulate this relationship are so diffuse that they too often appear ordinary, naturalized, and mundane, and therefore are rarely remarked on or analyzed. In order to make visible the function of computer graphics today, we must return to those early moments in the history of the technology when the gap between the physical world and its simulation is most clearly felt, when the theory of the world articulated by computer graphics was still in formation. For over two months Marsha Sutherland's Volkswagen occupied this space between worlds: an object in practice and an image in the making; one foot in the digital and another in Salt Lake; neither image nor object, but an image object trapped in an extended moment of becoming.[14] With a few clicks of my mouse, I can drop Marsha's Beetle into any modern graphical simulation, draping it in the newest texture and lighting algorithms, modeling its behavior as part of an interactive environment made from thousands of objects structured by this same logic (figure 0.3). Today the aerodynamic curve of all motor vehicles is the product of this transformation, a spline

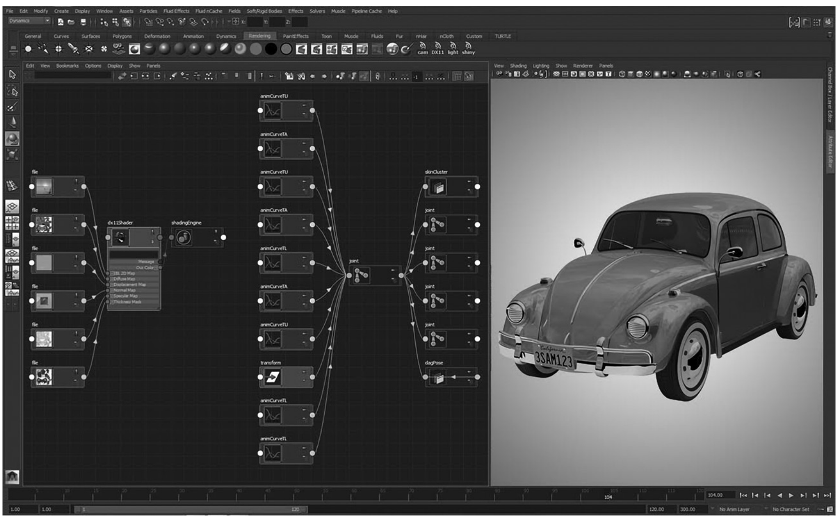

Figure 0.3
A contemporary VW Beetle simulation in Autodesk Maya. Note the visualization of the software's node architecture on the left, in which each of the elements that make up the rendered image (texture, lighting, geometry, etc.) are displayed as a nested structure of objects. Altered. Image by the author.

function driving around Salt Lake City, materially connected to that first digital object rendered out from Marsha's Volkswagen some fifty years ago. Ultimately this book is an effort to develop a language to speak to this quality of the world we now occupy. In doing so, we will find that computational images are not pictures of the things they represent; they are pictures of the world that produced them, and they execute a theory of that world in the world.

Visible Outputs

For over thirty years, computer graphics have been synonymous with illusion and artifice. Their appearance at the end of the twentieth century marked a crisis of visibility whereby the world was refigured as an image severed from the materiality of the thing it represents. Popular accounts of this transformation were commonplace in the enthusiasm surrounding new media technologies in the 1990s, a period often characterized by theories of the postmodern and the new, the supposed dissolution of the material into the virtual, and the rise of simulation across all facets of contemporary culture. This was also the period in which computer graphics first entered the realm of popular entertainment on a large scale, with the release of the first feature-length computer-animated film, broad success of Hollywood blockbusters that prominently featured computer-generated effects, and development of the first interactive 3D gaming consoles.[15] Along with the internet, computer graphics were one of the quintessential "new media" technologies of the decade. Just as scholars and critics touted the revolutionary power of the web, distributed networks, hypertext, and cyberspace, so too did they envision a future in which computer graphics would dominate our visual field, transforming our relationship to reality itself. As science fiction author Bruce Sterling exclaimed at the start of the decade, "The seams between reality and virtuality will be repeatedly and deliberately blurred. Ontology be damned!"[16]

Yet in many ways this moment was more aberration than innovation: a dramatic flourish of visibility that seemed to erupt fully formed before receding almost as quickly as it came. While today the internet continues to be viewed as one of the most important and pervasive technologies of our current media landscape, computer graphics seem an almost improper object whose vision of total simulation appears naive at best. Instead, the

past twenty years of media theory have seen a pronounced shift away from these immaterial preoccupations and toward the materiality of digital media as historically instantiated technical objects. New media, we are reminded, are not as new as they appear to be and have much in common with those older media forms they were said to replace.[17] If there *is* a radical transformation to be found at the heart of digital technologies, it lies in the procedural, algorithmic logic of computation itself, and not in the ways that computation is made meaningful to us through visualization. This materialist turn offers a valuable corrective to over a decade of enthusiastic writing on the transformative effect of the simulated image and of reading the rendered output of our machines with little regard for the means by which such images are made possible.[18] While the digital image was once thought to reveal the always already virtual nature of representation itself, under the material turn such images would seem to hide the truth of those technologies that ground all digital media, such that if we hope to understand the true function of our computers, we must look to the software, platforms, and code that structure them.[19]

This distinction implies a broadly hermeneutic critique whereby the machine conceals its function beneath the veneer of the digital image and its simulation, such that we are compelled to open the black box and look beyond mere representation.[20] Taken to the extreme, this formulation suggests that we have not simply misrecognized our true object of analysis, but have fallen victim to the illusory and seductive quality of digital images, which hide not only their material function as technical objects but likewise their role within the broader social and political circuits of computation. To imagine computational media as virtual or ephemeral erases the physical and affective labor required to build, maintain, and dismantle technical systems; their potentially catastrophic effects on the environment, human, and nonhuman life; and their political function in the lives of their users, the culture of their designers, and the shape of our societies.[21] As media scholar Tara McPherson has warned, "Our screens are cover stories, disguising deeply divided forms of both machine and human labor. We focus exclusively on them increasingly to our peril."[22]

Yet this wholesale refusal of the screen image produces its own restrictions. Despite this turn toward the mechanical interiority of technical things, our engagement with computing remains highly visual and deeply tied to the logic of simulation. It is true that our screens are not transparent windows

that lay bare the act of computing itself, but they are likewise not somehow outside that act, and play a principal role in shaping our understanding of and relationship to computational technologies. Yet in our rush to correct the visual bias of digital media studies, we have largely neglected the screen image as a material object in its own right—one with a heterogeneous history that runs parallel with that of textual or purely mathematical forms of computation. Rather than dismiss the visual as mere interface for deeper material processes, we might extend this materialist critique to include the simulated image, unpacking the means by which these images are modeled and displayed. Reading the digital image in this way—as an object structured by a set of distinct material practices—allows us to move beyond discourses of immateriality and virtuality to a theory of the digital image that is not visible in the rendered output of the screen. In doing so, we will find that computer graphics are one of the foundational technologies of our modern computational culture, and that they played a central role in the development of computing over the past seventy years.

To begin, we must unlearn the way we look at computational images. It does not seem controversial to suggest that our visual and material landscape has been fundamentally transformed by computation, yet this quality often cannot be deduced simply by looking.[23] Popular discourses of realism and fidelity dominate our analyses of digital image technologies, but are derived from an uncritical appropriation of those formal qualities that have historically defined prior modes of image making. As countless scholars have argued, digital images do not hold an indexical relationship to the world they represent, such that to analyze them exclusively in terms of their ability to reproduce the aesthetics of film and photography is to willfully occlude the means by which they are produced. This does not mean we must ignore the visual altogether. Rather, we must attend to the Janus-faced nature of digital images, which are shaped not by the etching of light but by the articulation of a set of computational objects developed to enact this simulation. Computer graphics exist simultaneously as both an assemblage of technical objects and an image that has been rendered out from them. To examine computational images in isolation from these objects is to mistake the render for the thing itself, and be drawn into an uncritical and ahistorical relationship that makes one culpable in the forms of material erasure so widely critiqued by media scholars today. If we wish to understand the function of these images, we must examine those objects

that surround them—objects that are the product of this distinct material history and articulate a distinctly computational ontology.[24]

Object Simulation

This connection between computer graphics and a computational theory of objects may seem counterintuitive. After all, "computer graphics" can be used to refer to any image produced by computer processing, from a single digital photograph to a fully interactive 3D environment. Not all graphical images are the product of object simulation in the sense that a digitized Volkswagen Beetle so clearly is. Nonetheless, nearly all contemporary computer graphics are structured by a theory of objects that emerged alongside these early experiments in the mid-twentieth century, in which the world is understood as a relational system of objects capable of discrete forms of interaction.

This distinction is visible in the first documented use of the term "computer graphics," formalized in 1960 by Verne Hudson, chief of preliminary design at the Wichita Division of the Boeing Airplane Company.[25] In 1964, a member of Hudson's team named William Fetter was the first person to model the human figure using a computer, crafting a three-dimensional object model out of vector lines that formed the shape of a sitting man (figure 0.4).[26] The figure appears as a transparent mesh of curves and angles, woven together with seven joints for basic movement and articulation. Its form was derived from United States Air Force anthropometric data, modeled by an engineer and transferred onto punch cards, and then fed into an IBM 7094 mainframe computer to produce a reel of magnetic tape that could be read by an automated plotting tool for paper output. The purpose of Fetter's model was to approximate the human body and provide adaptable representations for use in ergonomics and design. Its principal use was to model a pilot's ability to reach and grasp the various switches and dials found in the cockpit of the Boeing 747, designed from roughly 1964 to 1970 using a range of computer-aided techniques. The figure is commonly known among graphics researchers as "Boeing Man," and it is one of many origin stories in the history of computer graphics.[27] Fetter himself referred to the figure as First Man, implying a kind of archaeological lineage: the dawn of a new species form.[28]

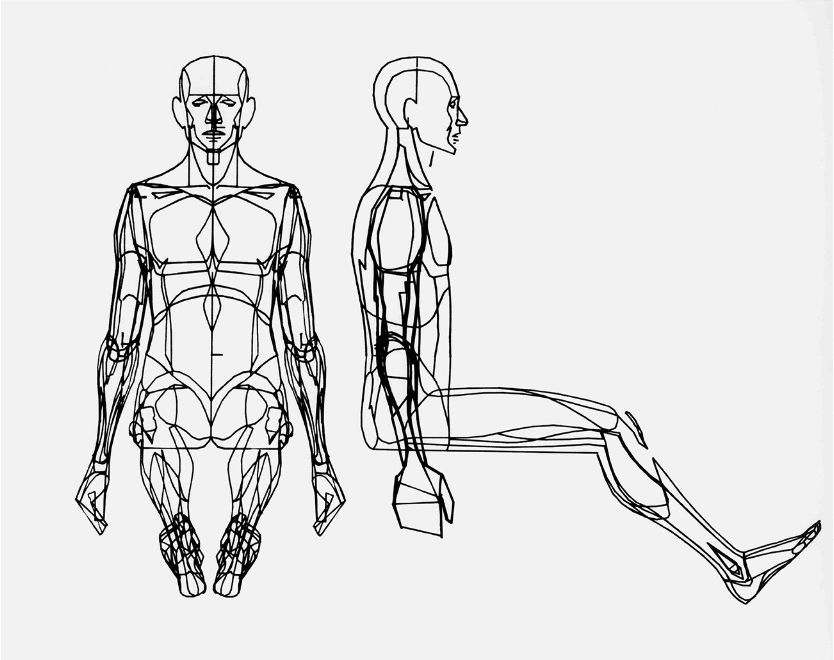

Figure 0.4
William Fetter, *First Man*, 1964. Courtesy of the Boeing Company.

Of course, Fetter's sitting figure is by no means the earliest example of what we now call computer graphics. Arguably the most visible graphical application in the history of early computing was the Semi-Automatic Ground Environment (SAGE) for air defense, commissioned and developed over the course of the 1950s after the US Air Defense Systems Engineering Committee recommended computerized networking for radar stations guarding the northern air approaches of the United States as a response to the threat of nuclear attack from the Soviet Union.[29] The SAGE system was a hugely ambitious sociotechnical apparatus made up of computers, network technology, radar, aircraft, and weaponry mobilized in the service of a global system. Designed to allow human operators to determine possible threats from long-range bombers, it required the complex cooperation of data transmission, calculation, and display. While SAGE was not exclusively or even primarily graphical, its visual interface was key to its operation. Using graphical consoles equipped with light guns, operators

Figure 0.5
Frame capture from IBM's short film "Freeing Man's Mind to Shape the Future" (1960), showing the graphical terminal of the Semi-Automatic Ground Environment air defense system.

could track two-dimensional representations of airplanes as they moved across a screen overlaid with a map of the part of the country under the defense of a given station (figure 0.5). When an operator identified a potential threat, the system would calculate an intercept path for fighter pilots or surface-to-air missiles before a decision was made whether or not to destroy the target.[30]

While SAGE was one of the earliest applications of large-scale interactive computer graphics, the image of the world that it articulates is fundamentally different than the one pictured by Fetter some ten years later. For SAGE, an enemy airplane is a blip on a screen, a target meant to be identified, part of a global system to be commanded (figure 0.6). Its visuality is two-dimensional and cartographic; its images functioned as symbols designed to elicit a response from a technical operator.[31] The SAGE system was a product of the Cold War environment that produced it, and articulated a

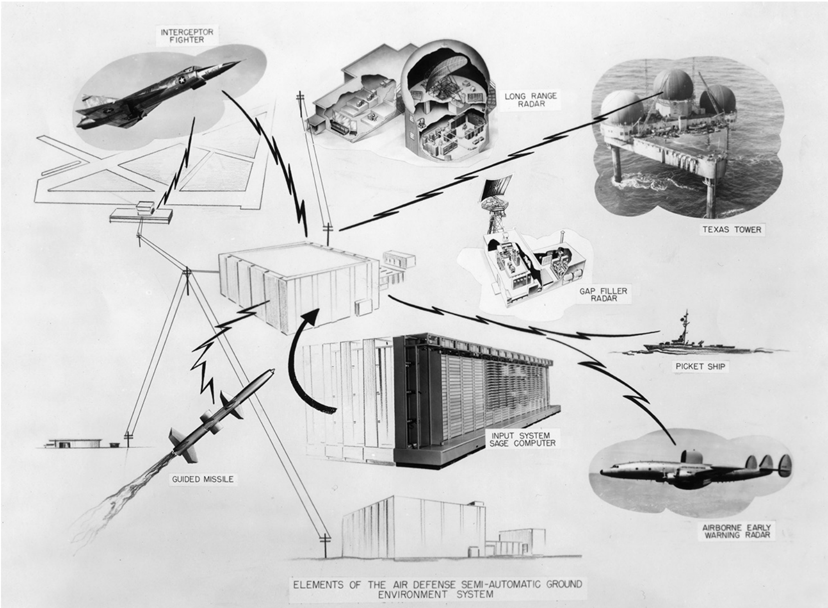

Figure 0.6
Diagram of the SAGE system as a complex sociotechnical apparatus. MITRE Corporate Records and Archives, SAGE collection, M0-139.

theory of that world as a system to be directed and controlled—a vision that would have long-standing repercussions for the development of communication technologies over the subsequent seventy years.[32] Fetter's airplanes are quite different. Here the plane forms the ground of a relational environment in which a human operator is situated (figure 0.7). This plane is not symbolic but rather mimetic, used to model or simulate a three-dimensional space comprised of a discrete set of interactive objects that includes this human figure, this First Man. The figure serves a standardizing function, its size and shape derived from what is called a 50 percentile figure, built to approximate the average size of 50 percent of air force pilots.[33] The shape of this model thus forms the basis for the design of a technical system—the 747 cockpit—and the assumption that its pilots' bodies will not vary widely from this presumptive norm.[34] In standardizing its human model and designing for that standardization, Boeing Man is shaped by a particular image of the world, and in turn comes to refigure the world

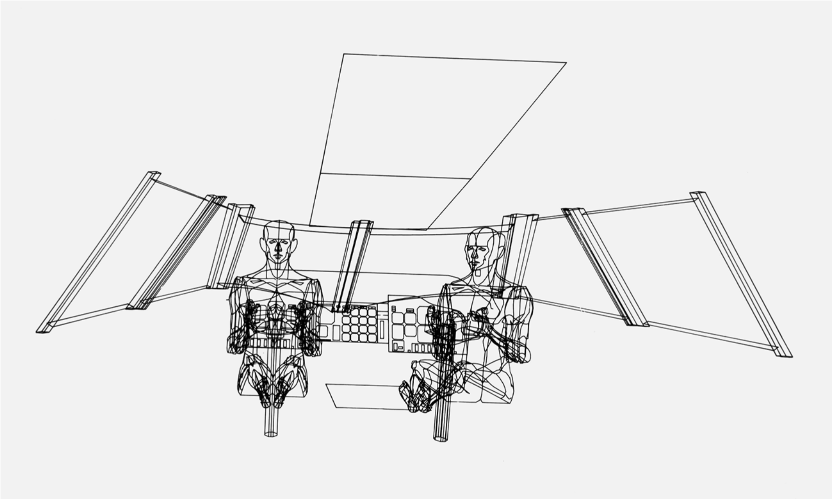

Figure 0.7
Fetter's Boeing Man as an interactive object within a simulated environment, 1964. Courtesy of the Boeing Company.

according to that image—a fact made evident in the thousands of Boeing 747 airplanes in operation today.[35] Understood this way, Fetter's image is a model for the primitive simulation of a physical object, designed to approximate and standardize the complexity of real-world interaction. It is as much a theory of the world as it is an image of it, and in the subsequent decades that theory would be made actionable.

In the ten years that separate these two moments, we find a pronounced transformation, and along with it a change in how computer graphics were understood to relate to the world that grounds them.[36] While the SAGE system treated graphics as *images* that visually represent numerical data to its operators, Fetter used graphics as a medium for the simulation of graphical *objects*. This is a subtle but deeply meaningful distinction, and one that is lost when we treat the history of computer graphics exclusively as a history of images produced through computational calculation.[37] It is a logic that emerges alongside computer graphics, growing to become altogether diffuse across the field of computer science as it begins to take up graphical systems and develop novel uses for computational images. As early as 1961, Ivan Sutherland was using object-oriented structures in the development of his widely influential Sketchpad program for computer-aided design

(CAD). Likewise, Steven Russel's *Spacewar!* (1962)—considered by many to be the first graphical, interactive digital game—was designed using object-oriented principles in this same period.[38] One of the earliest modern graphical user interfaces, developed from 1972 to 1979 at Xerox for use with the Alto computing system, was predicated on the object-oriented structure of the Smalltalk programming language to such an extent that the language is inextricably tied to its interface and requires it in order to function.[39] While each of these object forms cannot be made commensurate, as they do not all adhere to a single, fixed theory of object relation, they are nonetheless exemplary of a broad transformation in which object simulation becomes a principal structuring logic for computational systems.

In this way, the act of computing is refigured from a set of procedural calculations into an interactive environment, understood as a spatially embodied field of discrete computable objects. In short, computing is transformed from a *process* into a *medium*.[40] Today this object logic has grown into one of the dominant forms of our contemporary media environment, transforming the ways we model and represent the world, and in turn reorienting our understanding of that world as a structure of computable objects.[41] In exploring the transformation of computer science and its adoption of object simulation across a range of technical practices, this book proposes that to understand this reorientation, we must look to those sites from which it emerged, both as a moment in the history of computing and as an articulation of a distinct culture of practice.

Other Places

Just off the main campus of the University of Utah sits Fort Douglas, a military garrison founded in 1862 to protect the overland mail route and telegraph lines running from Salt Lake City to San Francisco. The site was strategically chosen in the foothills of the Salt Lake Valley, as the US military was concerned with secessionist activity in the area and wanted to keep an eye on the territory's Mormon population.[42] For nearly a century, the fort played a strategic role in the economic, social, and political stability of the region, but by the mid-1960s, much of the land had been transferred to the ownership of the university, and its buildings were frequently delegated for research projects run by Utah faculty and staff. It was in this context that in late 1968, an abandoned bunker in this former military garrison was

transformed into the home of one of the first commercial computer graphics firms in the United States (plate 2), known as the Evans and Sutherland Computer Corporation (E&S). In many ways the site exemplifies this early period in the development of computer graphics, with its proximity to military resources and isolation from the larger field of computer science. As is likely apparent, this was no place to start a computer hardware company, and for the first year researchers struggled to keep out dirt and drafts while working to maintain a stable electric grid. Yet this site marks the beginning of this strange history, if not the beginning of the computer graphics industry itself.

The 1960s were a transformative period in the history of computing. At the start of the decade, computation was still an expensive and highly limited resource, enabled by massive mainframes shared by dozens of researchers working asynchronously. Computing was a fundamentally noninteractive process: tasks needed to be programmed in advance onto physical media that could be submitted to a computer operator for calculation, and researchers would have to wait hours or even days for their calculations to be processed. These were industrial machines used for processing numerical data—more calculator than computer in any modern sense. Over the course of the decade this began to change, due in large part to the development of key technologies designed to interface human and machine.

The motivation for this shift was both technical and institutional, and involved the coordination of public funding with large-scale research initiatives driven by a strong vision for what the future of computing could be. In the United States, the principal player in this transformation was the Department of Defense and its Information Processing Techniques Office (IPTO), founded in 1962 and housed within the Advanced Research Projects Agency (ARPA).[43] Under the directorship of psychologist and computer scientist J. C. R. Licklider, the IPTO put forward a vision for the future of computing as a tool for "man-computer symbiosis," imagining a future in which "human brains and computing machines will be coupled together very tightly, [such] that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today."[44] Investing heavily in time-sharing, network technologies, artificial intelligence, and computer graphics, the IPTO pushed a vision of the computer as a device that would not only connect humans to one another but likewise connect human and

machine, allowing for new forms of communication and collaboration.[45] Far from the gatekeeping model of early mainframes, this new computer would be immediately accessible to individuals through real-time graphical interaction.[46]

It was in this context that David Evans was approached by University of Utah president James Fletcher to return to his alma mater in Salt Lake City and found a computer science division within the College of Engineering.[47] At the time Evans was an assistant professor at the University of California at Berkeley, having joined the College of Engineering in 1962 after a decade working in the computing division of the Bendix Corporation in Los Angeles. Evans was also a Salt Lake City native and received both his BS and PhD in physics from the University of Utah in the early 1950s.[48] At Berkeley, Evans had served as co–principal investigator for Project Genie—an early time-sharing system funded heavily by the IPTO—developing connections with government funders, and earning a reputation as a competent and effective research lead. Then in 1964, with the free speech movement erupting on the Berkeley campus, Evans made the decision to accept Fletcher's offer and return to Utah, taking with him a network of university and government connections that would be instrumental in establishing the Utah program.[49] The offer came with the full backing of the university to help shape a program in whatever way he saw fit, appointing him the director of computer science and computer operations in 1965 (figure 0.8).[50] Initial funds from the university were limited, but were supplemented by a $5 million grant from the IPTO that Evans was able to secure immediately following his hire. Paid out over the course of four years, the ARPA contract was devoted explicitly to "Graphical Man/Machine Communication," channeling Licklider's vision through the lens of graphical interaction.[51]

The program was deeply unconventional, recruiting graduate students that no other school would take, and fostering a kind of intellectual proving ground where students were encouraged to form their own collaborations with faculty and develop expert solutions that could be deployed broadly across multiple applications.[52] It is telling that despite the futurist aspirations that define much of today's culture of computing, many of these projects produced technologies that remain the de facto solutions for computer graphics, and are still widely used by researchers and artists today. Over the subsequent fifteen years, Utah became the epicenter of graphical research in the United States, attracting faculty from around the world and

a)

b)

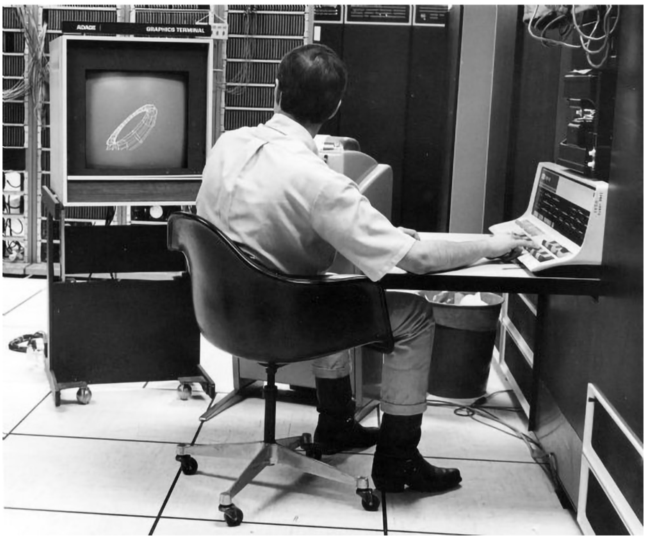

Figure 0.8

David C. Evans (top) and researcher in motorcycle boots (bottom) working in the University of Utah computing center, ca. 1968. Courtesy of the University of Utah School of Computing and the Special Collections Department, J. Willard Marriott Library, University of Utah.

launching the careers of dozens of researchers who would go on to define much of the commercial computing industry in the second half of the twentieth century.[53] In this sense, Utah served both as a test bed for early research that continues to shape the function of modern graphical systems, and as a network for early researchers who distributed that work to dozens of research programs as they moved out from Utah and into the emerging computing industry over the course of the 1970s and early 1980s.[54]

That the Utah program is at once so central to the history of computing and so absent from popular narratives of innovation reflects the contradictory role of computer graphics itself as a discipline within computer science. Even in this early period, graphics were considered by many to be a frivolous use of computing technology. Computational resources were a limited and extremely expensive commodity, and making pictures seemed to many a waste of time.[55] As several of its graduates recalled during a panel on the history of the Utah program at the ACM's SIGGRAPH conference in 1994,

> They knew that they were "onto something big" while outsiders at other universities disparaged the work in computer graphics as an illegitimate application of computing machinery. Computing research at that time involved computer languages, operating systems, and data processing. Graphics research required manipulating so much data to display images, that it pushed the envelope in computing technology.[56]

Computer graphics research was objectively impractical and unrealistic. The technologies that it required did not yet exist, and the computers themselves were not powerful enough to manipulate the massive amounts of data required for interactive graphical communication between a computer and user. Despite these challenges, IPTO directors viewed graphical interaction as central to the future of the field, and Evans was given the resources to develop the technologies to make these systems possible.[57] The Utah program benefited greatly from this hands-off approach, which by many accounts fostered a culture of research that operated largely independent of any broader consensus of what an appropriate object for computational research might be.[58]

By 1968, Evans had established Utah as a key research hub in an expanding network of ARPA-funded "centers for excellence" and looked to develop this work beyond the university by establishing a commercial venture. Evans had met Sutherland several years prior during his work on Project

Genie at Berkeley, and Sutherland later provided the initial ARPA funding for the Utah program during his two-year tenure as the IPTO's director. As the most prominent graphics researcher in the country and a close family friend, Sutherland was the obvious choice for a partner in a new commercial graphics venture, and while the Evans family initially planned to move to Cambridge, Massachusetts, to found the company in proximity to the funding and institutional partnerships of Boston's Route 128, ultimately it was Sutherland who moved to Salt Lake City in 1968 to cofound E&S in an abandoned military bunker just off the University of Utah campus.[59]

Plate 3 shows that same bunker five years later. The man on the left is Evans, and standing next to him is Shohei Takada of Hitachi Electronics. I found this photo inside a holiday greeting card sent in 1973 following a visit by Hitachi executives earlier that year to see the work being done in Salt Lake.[60] Taken together with plate 2, this image is emblematic of the dual role that Utah plays in the history of computing: at once isolated and experimental, yet simultaneously central, connected, and highly influential. Ultimately the same can be said of computer graphics. While making pictures with computers has been historically viewed as peripheral and inessential to the "real work" of computing, an examination of the history of computer graphics shows its key role in the growth of the modern computer, and along with it the transformation of our computational culture.

Image Objects

To understand this transformation, we must turn to those objects that enabled the emergence of computer graphics to begin with. Following this methodological imperative, each of the following chapters is structured around a distinct technical object: its history, the conditions of its emergence, its influence, and its afterlives. Through this object-oriented approach, I frame computer graphics as a structure of objects grounded in the historical conditions of their formation, but that continue to restrict and inform the ways we produce computational images today. To this end, the book follows a broadly chronological narrative, beginning with the earliest challenges of the then-nascent field of computer graphics at the start of the 1960s and focusing primarily on the role of the University of Utah as a cultural site from which the field is first articulated. Over time these clear distinctions will begin to dissolve, mirroring the historical transformation

of computer graphics as it grows across an ever-expanding range of technical disciplines and practices.

Chapter 1 explores early efforts to produce an algorithmic solution to the problem of visibility in a medium divorced from the physical restrictions of sight, optics, and light—what was known to computer graphics researchers as the *hidden surface problem*. In doing so, I critique attempts to fold computer graphics into a broad genealogy of the visual by mapping it onto existing techniques such as perspective projection, or the production of tricks and illusions, arguing instead that while computer-generated images offer the successful simulation of existing media forms, they construct vision in materially distinct ways. To examine the specificity of this construction, I look to early research into hidden surfaces for graphical display from 1963 to 1978, suggesting that the diverse and highly variable solutions to the problem of constructing visibility clearly separate computational vision from the optical regime of film and photography. Through the hidden surface problem, I contend that computer graphics are structured not by a logic of the visible but rather by processes whereby data are culled or erased such that the computer may more successfully interface with human vision. Here visibility becomes an algorithmic process of withholding whose specificity articulates a distinct theory of computer graphics as simultaneously screen image and simulated object—a tension that persists throughout this book and into the present.

In an effort to further distinguish computer graphics from the material specificity of those visual media they simulate, chapter 2 offers an analysis of memory and materiality through the history of the *computer screen* as a heterogeneous object that shifts and transforms in response to changes in the field of computer graphics from 1946 to 1975. Starting with the shift from calligraphic to raster graphics that begins in the late 1960s, I examine the affordances of early screens in order to identify those challenges that prevented computer graphics from adopting the scanline technology of early television displays. Ultimately the chapter identifies a single hardware object that structures and distinguishes the computer screen from other screen media: a piece of random-access computer memory for graphical display known as the frame buffer. This focus on the frame buffer introduces an additional set of questions around computer memory and its relationship to the visual image, the random as distinguished from the sequential, and memory as both a human and computational practice. The chapter

concludes by looking to the origins of the stored program concept along with the first experiments in computer graphics at MIT and Princeton in the late 1940s in order to make explicit this relationship between the screen and the random-access memory (RAM) of contemporary computing systems.

Having established the unique function of computer graphics as both visual representation and object simulation, chapter 3 explores the standardization of graphical objects in the mid-1970s, with an emphasis on questions of computational ontology. This period marks the moment in which computer graphics begins to actively digitize objects from the physical world, and in which new methods for simulating irregularity allowed for the creation of increasingly realistic images. From an examination of early techniques for the simulation of curved and shaded surfaces, I reflect on processes of standardization in computer graphics broadly, taking up perhaps the most famous graphical standard in the history of the field: an object known as the *Utah teapot*. Through an analysis of the teapot's history as a material object, research tool, and cultural practice, I look to identify how computer graphics understands, represents, and reproduces the world through simulation. Using the teapot as a foil, I ultimately argue for the materiality of simulated things and their wide-reaching influence beyond the field of computer graphics.

Moving through the 1970s, the strict focus on the University of Utah will begin to fall away as I follow the program's graduates and faculty as they enter the growing computer graphics industry. Likewise, the objects that make up the book's second half will become less representational, suggesting the diffusion of the structuring logic of computer graphics in ways that exceed its connection with the visual. Turning to language, chapter 4 argues for the primary role that graphics played in the reorientation of computer science toward the simulation of objects, with particular emphasis on the *object-oriented programming paradigm* developed by Alan Kay while a graduate student at the University of Utah in the late 1960s. Through an analysis of two early CAD systems, I demonstrate the influence of graphical paradigms on the structure of object-oriented systems generally. In doing so, I trace the afterlife of the Utah program through the circulation of this object logic in early graphical user interfaces and the rise of desktop publishing, documenting the history of the Adobe PostScript language in an aircraft carrier simulation built by E&S in the mid-1970s. In deploying

textual and linguistic objects, I suggest that computer graphics has had a structuring effect on the culture of computing that is not always legible as visual image, demonstrating the influence of the Utah program throughout the field of computer science in the second half of the twentieth century.

In examining the thirty-year prehistory of computer graphics, *Image Objects* ends where most histories begin. Arriving at the period in which computer graphics emerge in popular media over the course of the 1980s, chapter 5 pushes back against narratives that presume the inevitability of computer graphics' widespread adoption. Asking instead what technical and cultural shifts allowed for this rapid growth in the visibility of the medium, I suggest that the development of the *graphics processing unit* (GPU) by Utah graduate James Clark in the early 1980s allowed for the rapid proliferation of computer graphics across a range of applications and industries. In the GPU, each of the objects of the previous chapters is miniaturized and embedded within a single metaobject: a computer devoted exclusively to the task of graphical calculation. In this sense the GPU mirrors the object logic of this book itself, a metonym for the history of computer graphics as a whole. Tracking the emergence of the GPU as a set of conceptual shifts scattered throughout the history of computing, I assert the technology's principal contribution is its transformation of the algorithmic logic of software and memory into a physical object, formally fixing it for the purpose of acceleration and specialization. Ultimately the chapter argues that through the GPU, we can see an articulation of the historical claim of computing itself, whereby the complexity of computation as a cultural and historical practice is formalized and flattened through the crystallization of a procedural logic.

Computer graphics today are ubiquitous and invisible, as all manner of objects are produced, reshaped, and transformed by their encounter with computational images. Yet we have no language to describe this relationship, which exceeds the logical binaries we so often use to make sense of the world: the material versus the immaterial, the physical versus the digital, the natural versus the designed, the real versus the virtual. Each pair embeds different valences and represents different attempts to parse an image on a screen from a physical object in one's hand. And yet neither option sufficiently captures the tension inherent in the image object, which is neither material nor immaterial, neither natural nor designed, neither physical nor digital, but rather all of the above simultaneously. In

examining the world in this way, my hope is not to reveal some hidden ideology that sits beneath the veneer of the digital image, or propose some imaginary sense of the relation between the material and computational, but rather to make visible an operational continuity that stitches together distinctions we presume to be categorical yet have become coextensive under computation. Analyzing computer graphics in this way asks us to account for this and/both quality of our world, bringing it into being as an analytic and practical category in the hope that it might transform the ways we attend to how our world is articulated.